EVB2080 Evaluation Board
(EVB2080)
User's Manual

Freescale Semiconductor, Inc.

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and B are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative

The M•CORE name and logotype and the OnCE name are trademarks of Motorola, Inc.

© Motorola, Inc. 1999

CAUTION: ESD Protection

M•CORE development systems include open-construction printed circuit boards that contain static-sensitive components. These boards are subject to damage from electrostatic discharge (ESD). To prevent such damage, you must use static-safe work surfaces and grounding straps, as defined in ANSI/EOS/ESD S6.1 and ANSI/EOS/ESD S4.1. All handling of these boards must be in accordance with ANSI/EAI 625.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Contents

Section 1 Introduction

1.1 EVB2080 Features 9
1.2 System and User Requirements 10
1.3 EVB2080 Layout 10

Section 2 Configuration

2.1 Configuring Board Components 13
2.1.1 Setting the FLASH Data Bus Access Header (W1) 16
2.1.2 Setting the Chip Select Headers (W2, W4) 17
2.1.3 Setting the SRAM Data Bus Access Header (W3) 18
2.1.4 Using Processor Power Connect Headers (W5 — W8) 18
2.1.5 Setting the Software Select Switch (S2) 20
2.1.6 Setting the Memory Byte State Switch (S3) 21
2.2 Making Computer System Connections 22
2.3 Performing the EVB2080 Selftest 22
2.4 Memory Map 24
2.5 Using the Prototyping Area 25

Section 3 Operation

3.1 Debugging Embedded Code 27
3.1.1 Using the Picobug Monitor 27
3.1.2 Picobug Sample Session 29
3.1.3 Using the GNU Source-Level Debugger 32
3.2 Downloading to FLASH Memory 32
3.2.1 Using the SysDS Loader 32
3.2.2 Restoring System Software 37
3.3 MPB I/O Port Circuits 38
3.4 Accommodating Processor Differences 39

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Section 4 Connector Information

4.1 OnCE Connector (J27) 41
4.2 RS232 Connector (J26). 42
4.3 Prototyping Connector Sites (J14, J16) 43

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Figures

1-1 MMCEVB2080 Evaluation Board 11
3-1 SysDS Loader Main Screen 33
3-2 Upload To File Dialog Box 34
3-3 Display Flash/Ram Display 35
4-1 OnCE Connector J27 Pin Assignments 41
4-2 Connector Location J14 Pin Assignments 43
4-3 Connector Location J16 Pin Assignments 45

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Tables

1-1 MMCEVB2080 Evaluation Board Specifications 12
2-1 Component Configuration Settings 13
2-2 S2 Subswitch Settings. 20
2-3 Reset Vector Source 21
2-4 EVB2080 Self-Test LED Sequence 23
2-5 EVB2080 Default Memory Map 24
3-1 Picobug Commands 28
3-2 Memory Range/Sector Contents. 37
3-3 MPB, MPA Connections 39
4-1 OnCE Connector J27 Signal Descriptions 41
4-2 RS232 Connector J26 Pin Assignments 42
4-3 Connector Location J14 Signal Descriptions 43
4-4 Connector Location J16 Signal Descriptions 45

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Section 1 Introduction

This user's manual explains connection, configuration, and operation information for the MMCEVB2080 Evaluation Board (EVB2080), a development tool of Motorola's M•CORE™ family. The EVB2080 lets you develop code to be embedded in either an MMC2080 or MMC2075 microcontroller unit.

A standalone tool, the EVB2080 uses an RS232 connection to your computer. This connection lets you use Motorola's M•CORE System Development Software (SysDS) or the GNU source-level debugger. The SysDS consists of a loader, the Picobug monitor, the ESL monitor, and the built-in selftest. The EVB2080 also has a OnCE™ connector, enabling you to use a debugging application that requires one.

Optionally, you may use the EVB2080 with a different emulator product, such as the Motorola Embedded Background Debug Interface (EBDI).

Motorola's SysDS loader lets you download your code into the EVB2080's SRAM (for execution) or FLASH memory (for execution or for storage in non-volatile memory). Should your application overwrite factory programming in the FLASH memory device, you can use the SysDS loader to restore the factory programming.

1.1 EVB2080 Features

The EVB2080 features:

- MMC2080 resident MCU (a development version, in a 208-pin PQFP package).
- 1 megabyte FLASH memory.
- 512 kilobytes 70-nanosecond SRAM.
- Power supply that converts line power to 12-volt input power.
- An RS232 serial communication port.
- Designed for a 16-bit data bus, but can be configured for 8-bit bus operation.
- OnCE connector.
- 4 user-accessible LEDs.
- 4 general-purpose DIP switches (2 shared with power-up software selection).
- User prototyping (breadboard) area.
- Provision for user installation of optional prototyping connectors.
- Motorola's System Development Software (SysDS).
- GNU source-level debugger (from the Free Software Foundation).

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

1.2 System and User Requirements

You need an IBM PC or compatible computer, running the Windows 95 or WindowsNT (version 4.0) operating system. The computer requires a Pentium (or equivalent) microprocessor, 16 megabytes of RAM, 50 megabytes of free hard-disk space, an SVGA color monitor, and an RS232 serial-communications port. To use the Picobug debug monitor, you also need Hyperterminal or a comparable terminal-emulation program.

To get the most from your EVB2080, you should be an experienced C or M•CORE assembly programmer.

Your EVB2080 requires 12-volt input power, at 0.5 amperes. The power supply that comes with your EVB2080 provides this voltage from line power.

1.3 EVB2080 Layout

Figure 1-1 shows the layout of the EVB2080. Jumper header W1 specifies 8-bit or 16-bit FLASH data-bus access. Jumper header W2 specifies the FLASH chip select. Jumper header W3 specifies 8-bit or 16-bit SRAM addressing. Jumper header W4 specifies the SRAM chip select. Jumper headers W5 through W8 are power-interrupt headers, available for current measurement.

Connector J19 is the connector for 12-volt input power. Connector J26 is the RS232 serial connector. Connector J27 is the OnCE connector.

Note the unpopulated locations for optional, user-supplied connectors: J14 and J16, for connection to M2080 general-purpose I/O and other system signals; and J15, for a jumper header that grounds the processor pullup-enable signal.

Switch S1 is the reset switch. Switch S2 is the general-purpose configuration switch, which specifies the software module to be run upon reset. Switch S3 configures several aspects of memory operation, including 16- or 8-bit operation. Switch S4 simulates a low-battery condition at reset. Switch S5 is the power switch.

Location F1 is for the EVB2080 fuse.

LEDs DS1 through DS4 (also designated STAT3 through STAT0, respectively) are general-purpose status indicators. LED DS5 confirms 3.3-volt power (and therefore that all parts of the board are powered).

The EVB2080 prototyping area is between switch S4 and jumper header W8.

The resident MCU, at location U12, is a PMC2080PW001 device, which simulates an MMC2080 device.

Freescale Semiconductor, Inc.
For More Information On This Product,
Go to: www.freescale.com

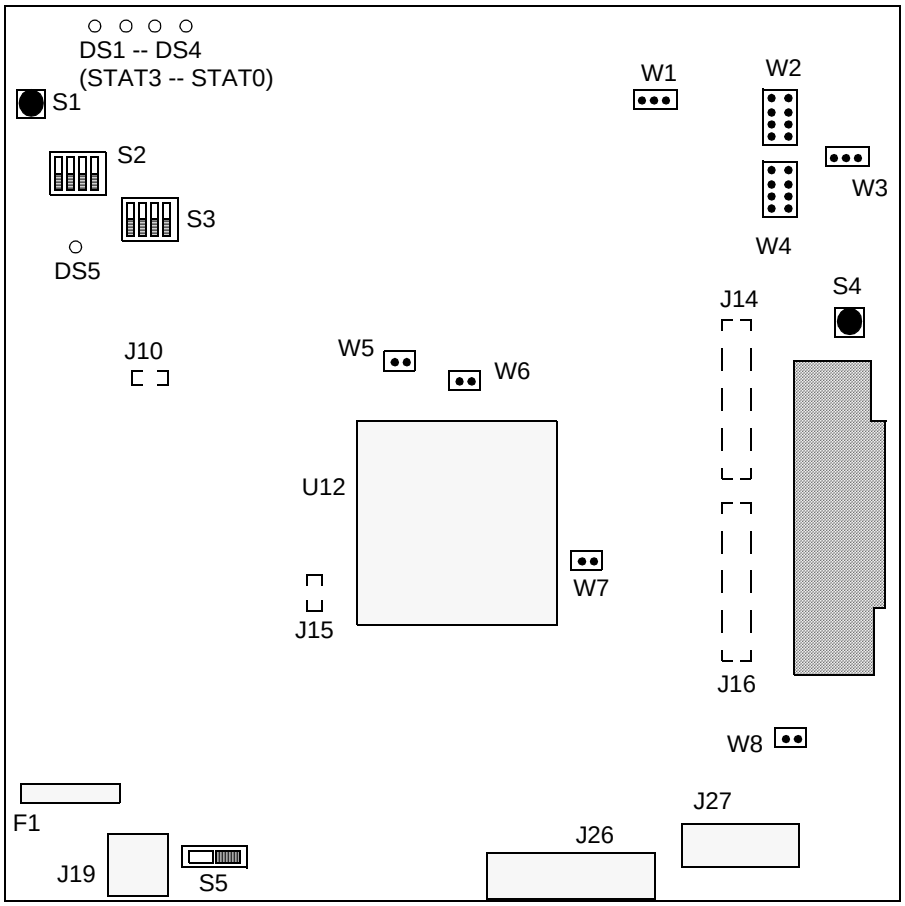


Figure 1-1 MMCEVB2080 Evaluation Board

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 1-1 lists EVB2080 specifications.

Table 1-1 MMCEVB2080 Evaluation Board Specifications

Characteristic	Specifications
MCU extension I/O port	HCMOS compatible
Operating temperature	0° to 40° C
Storage temperature	-40° to +85° C
Relative humidity	0 to 90% (non-condensing)
Internal clock	20 megahertz, set by system software, via the PLL function of the resident MCU.
External clock	10 megahertz, set by system software, via the PLL function of the resident MCU.
On-board crystal	160 kilohertz
Power requirements	12 volts dc, at 0.5 amperes, provided from a separate power source
Dimensions	6.9 x 8.2 inches (175 x 208 mm)

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

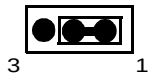
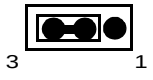
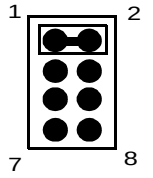
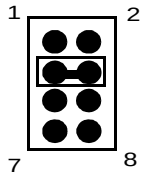
Section 2 Configuration

This section explains how to configure your EVB2080, and how to hook it up to your computer system.

2.1 Configuring Board Components

Table 2-1 is a summary of configuration settings; subsections 2.1.1 through 2.1.6 give additional information.

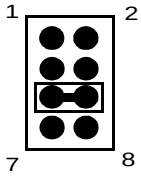
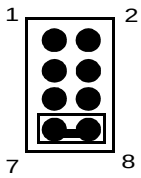
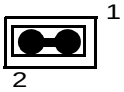
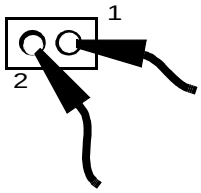
Table 2-1 Component Configuration Settings

Component	Position	Effect
FLASH Data-Bus Access Jumper Header, W1		Configures 16-bit accesses. Mandatory for using SysDS. (Subswitches S3-3 and S3-4 also must be set for 16-bit mode. Header W3 can configure either 16-bit or 8-bit mode for SRAM.) **Factory setting.**
		Configures 8-bit accesses. (Subswitches S3-3 and S3-4 also must be set for 8-bit mode. Header W3 must configure 8-bit mode for SRAM.)
FLASH Chip Select Header, W2, or SRAM Chip Select Header, W4 (Each header should have just one jumper.)		Configures chip select 0. **Factory setting for header W2.** (Required W2 configuration for SysDS.)
		Configures chip select 1. **Factory setting for header W4.** (Required W4 configuration for SysDS.)

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

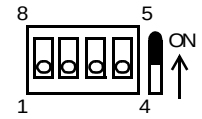
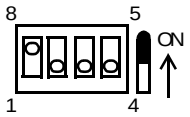
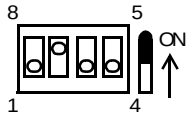
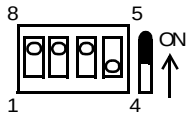
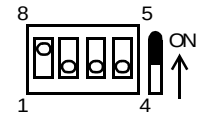
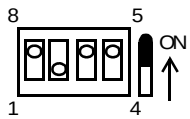
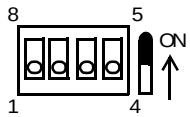
Table 2-1 Component Configuration Settings

Component	Position	Effect
FLASH Chip Select Header, W2, or SRAM Chip Select Header, W4 (continued)		Configures chip select 2.
		Configures chip select 3.
SRAM Data-Bus Access Jumper Header, W3		Configures 16-bit accesses. Mandatory for using SysDS. (Allowed only if header W1 configures 16-bit mode for FLASH.) **Factory setting.**
		Configures 8-bit accesses. (Mandatory if header W1 configures 8-bit mode for FLASH.)
Processor Power Connect Headers, W5 — W8		Applies power to processor modules: W5 — oscillator, W6 — PLL, W7 — core; W8 — I/O. **Factory setting.**
		Measures power to the processor module.
Reset Switch, S1		Push to reset all board components.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

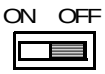
Table 2-1 Component Configuration Settings

Component	Position	Effect
Software Select Switch, S2		Runs user code starting at address 0x0102_0000.
		Runs Picobug monitor for software development. **Factory setting.**
		Runs ESL monitor for software development.
		Runs built-in self-test, which can detect board-level system problems.
Memory Byte State Switch, S3		Specifies external boot source, reset vector from the external peripheral CS0 specifies, and 16-bit FLASH bus accesses. (One of 3 plausible configurations.) **Factory setting.**
		Specifies external boot source, reset vector from the external peripheral CS0 specifies, and 8-bit memory bus access. (Another of 3 plausible configurations.)
		Specifies reset vector from internal ROM and 16-bit memory bus access. (Last of 3 plausible configurations.)
Low Battery Switch, S4		Push to simulate a low-battery condition. (This pulls high the processor LOBAT pin.)

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 2-1 Component Configuration Settings

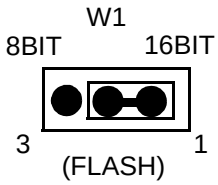
Component	Position	Effect
Power Switch, S5		Turns ON EVB2080 power (+5 volts and +3.3 volts).
		Turns OFF EVB2080 power (+5 volts and +3.3 volts — 12-volt power remains present). **Factory setting.**

2.1.1 Setting the FLASH Data Bus Access Header (W1)

Jumper header W1 specifies the width of FLASH data-bus accesses. The diagram below shows the factory configuration: the jumper between pins 1 and 2 specifies 16-bit accesses. (This W1 configuration connects the most significant bit of the 16-bit bus to FLASH memory.)

NOTES: *You must use 16-bit mode in order to use Motorola System Development Software (SysDS).*

If header W1 configures 16-bit mode, S3 subswitches 3 and 4 also must be set for 16-bit operation. Header W3 can configure either 16-bit mode or 8-bit mode for SRAM accesses.



Alternatively, you may specify 8-bit data-bus accesses by repositioning the W1 jumper between pins 2 and 3. (This W1 configuration connects the processor address bit 0 to FLASH data bit 15.)

NOTE: *If header W1 configures 8-bit mode, S3 subswitches 3 and 4 also must be set for 8-bit operation. Header W3 must configure 8-bit mode for SRAM accesses.*

Although less efficient than 16-bit mode, 8-bit mode is appropriate for some situations. For example, it makes additional general-purpose I/O lines available to your application. But 8-bit mode prevents access to half the EVB2080 RAM, and prevents you from using SysDS. Furthermore, you must configure processor internal registers correctly. (For details, see the processor user's guide.)

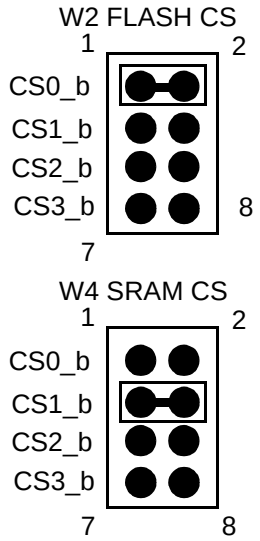
Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

2.1.2 Setting the Chip Select Headers (W2, W4)

Jumper header W2 specifies the FLASH chip select; jumper header W4 specifies the SRAM chip select. The diagram below shows the factory configuration. The jumper between W2 pins 1 and 2 specifies chip select 0 for FLASH memory. The jumper between W4 pins 3 and 4 specifies chip select 1 for SRAM.

NOTE: *You must use the factory settings of the diagram in order to use Motorola System Development Software (SysDS).*



Alternatively, you can select chip select 1, 2, or 3 for FLASH memory by repositioning the W2 jumper, respectively, to pins 3 and 4, 5 and 6, or 7 and 8. Similarly, you can select chip select 0, 2, or 3 for SRAM by repositioning the W4 jumper, respectively, to pins 1 and 2, 5 and 6, or 7 and 8.

NOTES: *1. FLASH memory and SRAM must have different chip selects: the configurations of headers W2 and W4 must not match. Do not install multiple jumpers in either header.*

2. You must have W2 specify chip select 1, and W4 specify chip select 0, as part of restoring system software to FLASH memory sectors 0 through 4. Subsection 3.2.2 gives complete instructions for restoring these sectors.

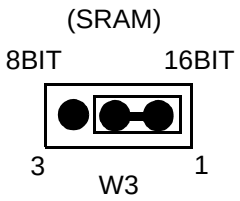
Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

2.1.3 Setting the SRAM Data Bus Access Header (W3)

Jumper header W3 specifies the width of SRAM data-bus accesses. The diagram below shows the factory configuration: the jumper between pins 1 and 2 specifies normal, 16-bit mode. (This W3 configuration connects the processor's next sequential address bit to the most significant bit of RAM. That is, SRAM connects to processor address bits A18 through A1.)

NOTE: *You must use 16-bit mode in order to use Motorola System Development Software (SysDS).*



Alternatively, you may specify 8-bit data-bus accesses by repositioning the W3 jumper between pins 2 and 3. (This W3 configuration connects the processor least-significant address bit of RAM. That is, SRAM connects to processor address bits A17 through A0.)

Although less efficient than 16-bit mode, 8-bit mode is appropriate for some situations. For example, it makes additional general-purpose I/O lines available to your application. But 8-bit mode foregoes access to half the EVB2080 RAM, and prevents you from using SysDS. Furthermore, you must configure processor internal registers correctly. (For details, see the processor user's guide.)

NOTE: *If header W1 configures 8-bit mode, header W3 also must configure 8-bit mode.*

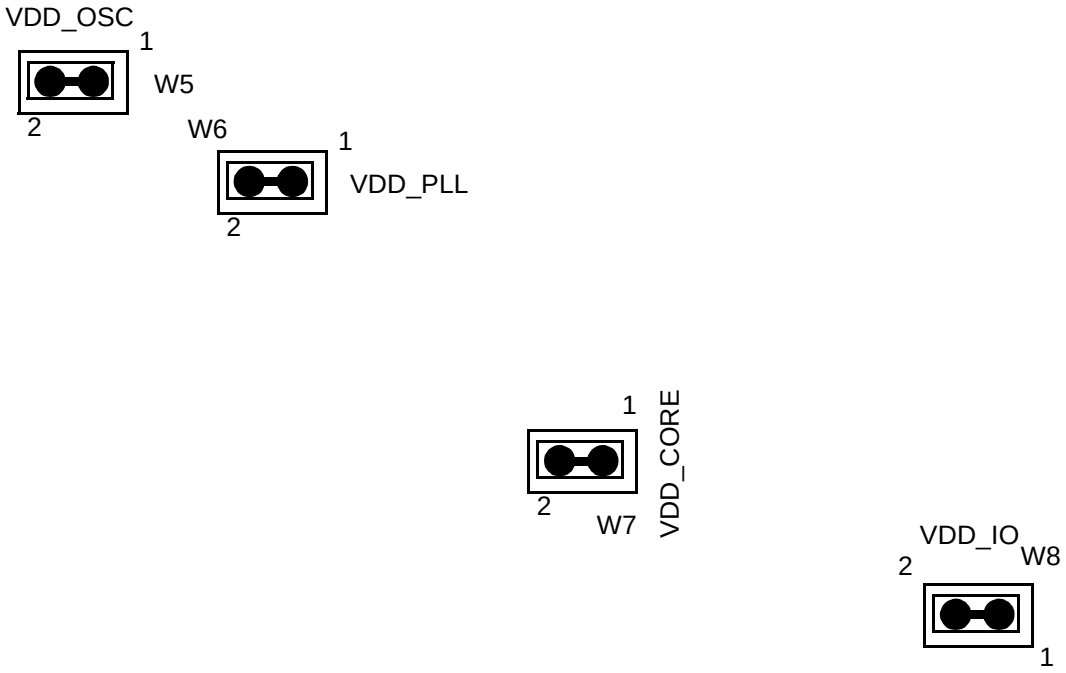
2.1.4 Using Processor Power Connect Headers (W5 — W8)

Jumper headers W5 through W8 connect 3.3-volt power to specific modules of the resident MCU:

- W5: Oscillator module
- W6: PLL module
- W7: Core module
- W8: I/O module

Freescale Semiconductor, Inc.
For More Information On This Product,
Go to: www.freescale.com

The diagram below shows the factory configuration for all four of these headers: the jumper between pins 1 and 2 connects power to the corresponding module.



NOTE: *Unless you are measuring power, leave jumpers installed in headers W5 through W8.*

To measure the current going to a particular module,

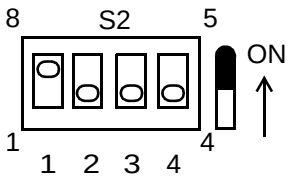
1. Turn off board power. (Turn OFF switch S5 or disconnect the power supply.)
2. Remove the jumper from the corresponding header.
3. Connect an ammeter to the header pins.
4. Restore board power and measure module current.
5. When your measurement is done, turn off board power.
6. Disconnect the ammeter from the header pins; replace the jumper.
7. Restore board power.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

2.1.5 Setting the Software Select Switch (S2)

Switch S2 specifies the software to be run on the processor upon a reset. The diagram below shows the factory configuration: subswitch 1 in the ON position and subswitch 2 in the OFF position. (The positions of the remaining subswitches does not matter, unless you configure the built-in selftest. Subsection 2.3 explains the selftest.) This specifies the Picobug monitor.



To specify a different software module, reset the S2 subswitches per **Table 2-2**.

Table 2-2 S2 Subswitch Settings

Software Module	Subswitch 1	Subswitch 2
Built-in selftest	ON	ON
Picobug monitor	ON	OFF
ESL monitor	OFF	ON
User code	OFF	OFF

NOTES: *1. The Picobug and ESL monitors include a brief power-on selftest of SRAM. As such software begins execution, the four yellow LEDs, DS1 through DS4 flash on, then go off. This means that the SRAM passed the test, and the software continues execution.*

If the LEDs remain on, it means that the SRAM failed the test. You should contact Motorola customer support for assistance.

2. ESL monitor requires additional debug software on your computer for compatibility with your EVB2080.

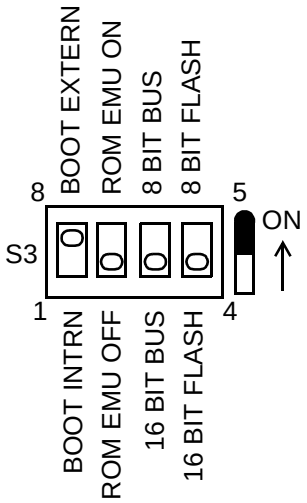
3. You also may use switch S2 for control of your own application software. Subsection 3.3 explains this additional role.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

2.1.6 Setting the Memory Byte State Switch (S3)

Switch S3 configures several aspects of memory. The diagram below shows the factory configuration: boot from an external source, 16-bit data bus, and 16-bit FLASH operation.



To change such memory aspects, reposition the corresponding subswitches per this guidance:

- Subswitch 1 (pins 1 and 8) selects the internal or external boot source. As **Table 2-3** explains, the position of subswitch 1 also determines the source of the reset vector.
- Subswitch 2 (pins 2 and 7) must be OFF, as the EVB2080 does not support ROM emulation.
- Subswitch 3 (pins 3 and 6) and subswitch 4 (pins 4 and 5) *must* be set for the same size bus accesses that header W1 configures.

NOTES: *As part of the built-in selftest, you must set S3 subswitch 1 to BOOT EXTERN, and you must set S3 subswitch 2 to ROM EMU OFF. Subsection 2.3 explains how to run the selftest.*

You must set S3 subswitch 1 to BOOT INTERN, and S3 subswitch 2 to ROM EMU OFF, as part of restoring system software to FLASH memory sectors 0 through 4. Subsection 3.2.2 gives complete instructions for restoring these sectors.

Table 2-3 shows how S3 subswitch 1 affects the source of the reset vector.

Table 2-3 Reset Vector Source

S3-1 Position	Reset Vector Source
OFF	Internal ROM.
ON	CS0-connection external memory peripheral: usually, this is external FLASH.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

2.2 Making Computer System Connections

When you have configured your EVB2080, you are ready to connect it to your computer system:

1. Make sure that power is disconnected.
2. If you will use RS232 communication with your host computer, connect an RS232 cable between EVB2080 connector J26 and the appropriate serial port of your computer.
3. If you will use a OnCE-compatible emulator with your EVB2080, connect an appropriate 14-lead ribbon cable between EVB2080 connector J27 and your emulator. Then use an appropriate cable to connect your emulator to your host computer.
4. Set the EVB2080 power switch (S5) to OFF. Then connect your 12-volt power supply to line power and to EVB2080 connector J19. Set power switch S5 to ON: green LED DS5 lights to confirm that the EVB2080 is powered and converting input voltage.

   Should DS5 *not* light, you may need to replace the fuse at location F1, next to power connector J19. (Use a BUS GMA-1.5A fuse, or compatible.)
5. This completes system connections: you are ready to perform a selftest, per the instructions of subsection 2.3, below. You are ready to begin debugging or other development activities, per the instructions of Section 3.

2.3 Performing the EVB2080 Selftest

Once you have configured your EVB2080, you can perform a selftest of its components.

NOTE: *If you open Hyperterminal, per the instructions of subsection 3.1.1, Hyperterminal displays the progress of the selftest. Should the selftest fail, Hyperterminal indicates the address at which the test failed.*

1. Make sure that EVB2080 power is turned off or disconnected. The green power LED DS5 should be out.
2. Set switch S2 for the built-in selftest: subswitches 1, 2, and 3 ON.
3. Set switch S3 subswitch 1 to ON; set S3 subswitch 2 to OFF.
4. Turn on power. The green LED DS5 comes on to confirm power, and the EVB2080 begins its selftest.
5. Yellow LEDs DS1 and DS4 light during the write to SRAM. Then, during the read from SRAM, DS1 and DS4 go out, while DS2 and DS3 light. (Should all four LEDs stay lit at this point, they indicate that the SRAM test failed. In case of such a failure, which aborts the rest of the selftest, you should contact Motorola customer support for assistance.)

Freescale Semiconductor, Inc.
For More Information On This Product,
Go to: www.freescale.com

6. LEDs DS1 through DS4 light and go out during the test, according to the sequence of **Table 2-4**.

Table 2-4 EVB2080 Self-Test LED Sequence

DS1	DS2	DS3	DS4
OFF	ON	OFF	ON
ON	OFF	ON	OFF
ON	OFF	OFF	OFF
OFF	ON	OFF	OFF
OFF	OFF	ON	OFF
OFF	OFF	OFF	ON
ON	ON	ON	ON
OFF	OFF	OFF	OFF

7. When all four LEDs go out, the EVB2080 has passed the selftest. (If the LEDs stay lit, the EVB2080 has failed the selftest: contact Motorola customer support for assistance.)
8. This completes the self-test.
9. Turn off power.
10. Configure switch S2 for your next development activity before restoring power to the EVB2080.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

2.4 Memory Map

Table 2-5 is the default memory map for your EVB2080. Bold type indicates the factory settings. As subsection 2.1.2 specified, you must have different settings on jumper headers W2 and W4.

Table 2-5 EVB2080 Default Memory Map

Address Range	Chip Select	Jumper Setting	Memory Resource	Related Signal
0x0100_0000 **0x010F_FFFF**	0	**W2, pins 1,2**	**FLASH**	CS_b[0]
0x0100_0000 0x0107_FFFF		W4, pins 1,2	SRAM	
0x0200_0000 0x020F_FFFF	1	W2, pins 3,4	FLASH	CS_b[1]
0x0200_0000 **0x0207_FFFF**		**W4, pins 3,4**	**SRAM**	
0x0300_0000 0x030F_FFFF	2	W2, pins 5,6	FLASH	CS_b[2]
0x0300_0000 0x0307_FFFF		W4, pins 5,6	SRAM	
0x0400_0000 0x040F_FFFF	3	W2, pins 7,8	FLASH	CS_b[3]
0x0400_0000 0x0407_FFFF		W4, pins 7,8	SRAM	

Note that later memory ranges echo memory contents. For example, if the FLASH address range is 0x0100_0000 through 0x010F_FFFF, the same data is in successive ranges 0x0140_0000 through 0x014F_FFFF, 0x0180_0000 through 0x018F_FFFF, and 0x01C0_0000 through 0x01CF_FFFF.

Likewise, if the SRAM address range is 0x0200_0000 through 0x0207_FFFF, the same data is in successive half-megabyte ranges 0x0208_0000 through 0x020F_FFFF, 0x0240_0000 through 0x0247_FFFF, and so forth.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

2.5 Using the Prototyping Area

The EVB2080 prototyping area lets you add your own components to the board. Merely insert the component's feet through holes in the board; solder the feet in place to hold the component in position. Run appropriate leads from the new component to board power and ground locations.

Note that the top row of prototyping-area holes are +3.3-volt sources, the bottom row of holes are +5-volt sources, and the right partial column of holes are ground connections.

If you use an external prototyping board, consider installing your own Berg 69192-620 headers to locations J14 and J16. This would facilitate cabling GPIO, clock, reset, and power signals to external prototype circuits. Headers installed at J14 and J16 also would facilitate wire wrapping or probing. Section 4 includes pin assignments and signal descriptions for such optional headers.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Section 3 Operation

This section explains how to begin using debugging tools available for your MMCEVB2080 Evaluation Board, as well as how to use Motorola's SysDS Loader.

3.1 Debugging Embedded Code

With your EVB2080, you may use the Picobug monitor as standalone software. Optionally, you may use the GNU source-level debugger with the Picobug monitor. Other firms may produce still additional software to run, test, and modify the code you develop for embedding in an MMC2080 MCU.

3.1.1 Using the Picobug Monitor

The Picobug monitor comes burned into the FLASH memory device of your EVB2080**.** Before you start the Picobug monitor, make sure that you have an RS232 connection between EVB2080 connector J26 and a serial port of your computer.

To start the monitor, for use as a standalone debugger:

1. Make sure that power is *not* applied to your EVB2080.
2. Activate Hyperterminal or a comparable terminal-emulation program. (If you use a different terminal-emulation program, you must make corresponding changes in the commands and menu selections of these instructions, and in the instructions of paragraph 3.1.2.)
3. From the File menu, select Properties. This opens a properties dialog box.
4. Click on the Configure button of the dialog box. This opens a configuration dialog box.
5. Use the configuration dialog box to make these communications settings: 19,200 baud, 8 bits, no parity, 1 stop bit, and no flow control. Also set the correct communications port (for example, COM1). Click the OK button of the dialog box.
6. Set switch S2 for the Picobug monitor: subswitch 1 ON and subswitch 2 OFF.
7. Apply power to the EVB2080 and press the enter key. The Picobug monitor starts automatically, displaying the command prompt: `picobug>`.

To use the Picobug monitor, merely enter commands at the prompt. **Table 3-1** explains these commands. To see a list of these commands on your computer screen, enter a question mark or the extra command `he` at the command prompt.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 3-1 Picobug Commands

Command	Explanation
baud [*value*]	Set Baud Rate: • With optional *value* value, sets that rate (300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200). • Without any *value* value, sets default rate: 19200 baud.
br [*address*]	Breakpoint: • With optional *address* value, sets a new breakpoint at that address. • Without any *address* value, lists all current breakpoints.
g [*address*]	Go: • With optional *address* value, starts code execution from that address. • Without any *address* value, starts code execution from the current program-counter value. In either case, execution stops when it arrives at a breakpoint.
gr	Go to Return: Executes code from the current program-counter value to the return address of the calling routine. (Should execution arrive at a breakpoint before encountering the return address, execution stops at the breakpoint.)
gt *address*	Go to Address: Executes code from the current program-counter value to the specified *address* value. (Should execution arrive at a breakpoint before encountering the specified address, execution stops at the breakpoint.)
he	Help Displays available commands, identical to the ? command.
lo [*address*]	Download: • With optional *address* value, downloads a binary image to that address in SRAM. • Without any *address* value, downloads to SRAM an S-record text file.
md [*address1* [*address2*]] [;*size*]	Memory Display: • With optional *address1* and *address2* values, displays memory contents between the addresses. • With optional *address1* value, displays contents of 16 memory bytes. • With no address value, defaults to the last address viewed. • The optional *size* value specifies the format: **b** (bytes, the default), **h** (half words), **w** (words), or **i** (instructions).
mds [*address*]	Memory Display 256: • With optional *address* value, displays contents of 256 memory bytes, starting at that address. • With no address value, displays contents of 256 memory bytes, starting from the last address viewed.
mm [*address* [*value*]] [;*size*]	Modify Memory: • With optional *address* and *value* parameter values, assigns that value to the *address* location. • With optional *address* value but no *value* parameter value, prompts for a value for the *address* location, then prompts for a new value for the next location. To stop modification, enter a period instead of a new value. • With no optional *address* value, prompts for a value for the last address viewed, then prompts for a new value for the next location. To stop modification, enter a period instead of a new value. • The optional *size* value, specifies the format: **b** (bytes, the default), **h** (half words), **w** (words), or **i** (instructions).

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 3-1 Picobug Commands (Continued)

Command	Explanation
nobr [*address*]	No Breakpoint: • With optional *address* value, removes the breakpoint from that address. • Without any *address* value, removes all the breakpoints.
reset	Reset: Resets the CPU and peripherals.
rd [*name*]	Register Display: • With optional *name* value, displays the value of that CPU register. • Without any *name* value, displays the values of all CPU registers.
rm *name value*	Register Modify: Assigns the *value* parameter value to the *name* CPU register.
t	Trace (Step): Single steps one instruction; identical to the s command.
s	Step (Trace): Single steps one instruction; identical to the t command.
?	Help Displays available commands, identical to the he command.

3.1.2 Picobug Sample Session

1. This sample session begins with the Picobug prompt:

```
picobug
```

2. To see the contents of all registers, enter the Register Display (rd) command without any name value:

```
picobug>     rd
```

   The system responds with a display such as this:

```
     pc 02000286   epc       02000286   fpc       0010a000
    psr 80000100   epsr      80000100   fpsr      00020000
ss0-ss4 bad0beef   20000c00  20008000   20010042  00000801   vbr       02005c00
  r0-r7 020027f8   00000050  0000ea60   02005f94  02006708   80070101  00000200  00000040
 r8-r15 0010a000   00020000  20000c00   020067c0  00000000   02005f94  10005000  02000286
```

3. To see the contents of a specific register, such as the epc register, enter the Register Display (rd) command *with* the name value:

```
picobug>  rd epc
```

   The system responds with a display such as this:

```
epc:      02000286
```


For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

4. To see the contents of a specific memory location, enter the Memory Display (md) command with the location address. An optional size value (in this case w, for word) may be part of the command:

```
picobug>  md    0x02001000   ;w
```

   The system responds with a display such as this:

```
02001000:  8EF0B37E
```

5. To see the contents of a memory range, enter the Memory Display (md) command with the beginning and ending addresses. An optional size value (in this case b, for byte) may be part of the command:

```
picobug>  md    0x02000000   0x02000010   ;b
```

   The system responds with a display such as this:

```
02000000: 8E  F0  00  00  55  55  55  55  0E  22  9E  E8  03  20  0D  20  0E $..UUUU.  "…  . .
02000010: F7
```

6. To download into SRAM a program executable, in S-record format, enter the Download (lo) command without any address value:

```
picobug>  lo
```

   The system waits for you to send the program executable file. To do so, open the Transfer menu and select Send Text File. This opens a file-select dialog box. Use this dialog box to specify the appropriate S-record file, then click on the Open button. As soon as the download is complete (this may take several minutes), the Picobug prompt reappears:

```
picobug>
```

7. To see the new contents of registers, enter the Register Display (rd) command again, without any name value:

```
picobug>  rd
```

   The system responds with an updated display, which shows that the pc register value reflects the start of the program just downloaded:

```
    pc 0200022a   epc       2d00108a   fpc       0010a000
   psr 80000000   epsr      80070101   fpsr      00020000
ss0-ss4 bad0beef   20000c00  20008000   20010042  00000801   vbr       02005c00
  r0-r7 bad0beef   00000050  00000000   d89f69ab  02005f20   80000000  00000200  00000024
 r8-r15 0010a000   020066b8  020067d7   02006948  02006714    02006708  020067c8 2d0001c4
```

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

8. To set a breakpoint at address 0x0200025C, enter this address as part of the Breakpoint (br) command:

```
picobug>  br    0x0200025c
```

   The Picobug prompt reappears, confirming that the system set the breakpoint:

```
picobug>
```

9. To see the list of breakpoints, enter the Breakpoint (br) command *without* any address value:

```
picobug>  br
```

   The system responds with the addresses of breakpoints, in this case only the breakpoint set in step 8:

```
0200025C
```

10. To start program execution, enter the Go (g) command:

```
picobug>  g
```

    In this instance, the breakpoint set during step 8 stops code execution. The system responds with this new display of register values:

```
  At breakpoint!!
     pc 0200025c   epc       0200025c   fpc       0010a000
    psr 80000100   epsr      80000100   fpsr      00020000
ss0-ss4 bad0beef   20000c00  20008000   20010042  00000801   vbr       02005c00
  r0-r7 020027f8   00000050  0000ea60   d89f69ab    02005f20 80000000  00000200  00000040
 r8-r15 0010a000   020066b8  020067d7   02006948  02006714   02006708  10005010  020002a2
```

11. To remove all breakpoints, enter the No Breakpoint (nobr) command, without any address value:

```
picobug>  nobr
```

    The Picobug prompt reappears, confirming that the system has removed the breakpoints:

```
picobug>
```

12. To see the list of breakpoints again, once more enter the Breakpoint (br) command without any address value:

```
picobug>  br
```

    As there are no longer any breakpoints, the system responds with the Picobug prompt:

```
picobug>
```

13. To continue with this example session, enter another appropriate command. For example, to resume program execution, enter the Go (g) command.

14. To end your Picobug session, remove power from the EVB and close the terminal-emulation program.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

3.1.3 Using the GNU Source-Level Debugger

The GNU source-level debugger is on the CD-ROM that comes with your EVB2080. This GNU software works with the Picobug monitor to provide source-level debugging for your code.

The EVB2080 software release guide gives the instructions for loading the GNU software. The EVB2080 example explained in the quickstart document makes use of the GNU debugger.

3.2 Downloading to FLASH Memory

The Motorola SysDS Loader lets you program code into FLASH memory, upload FLASH contents to a PC file, verify that FLASH contents match those of a download file, display memory contents, erase FLASH memory, erase a sector of FLASH memory, or blank check a sector of FLASH memory.

3.2.1 Using the SysDS Loader

Follow these steps to use the SysDS Loader:

1. If you have not already installed the SysDS Loader onto your computer hard disk, do so. The EVB2080 product release guide includes installation instructions.
2. If the Hyperterminal emulation program is running, stop the program. (The SysDS Loader needs the same computer serial port that Hyperterminal uses.
3. Set switch S2 for the Picobug monitor: Set subswitch 1 ON and subswitch 2 OFF.
4. Press switch S1 to reset the EVB2080.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

5. Start the SysDS Loader. The main screen (**Figure 3-1**) appears.

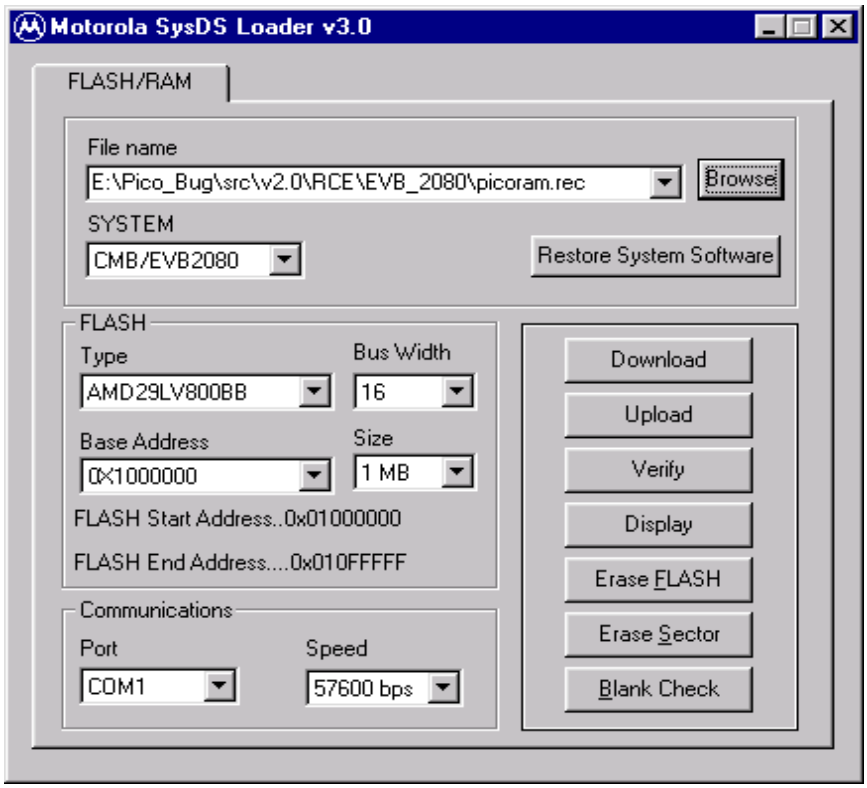


Figure 3-1 SysDS Loader Main Screen

6. Go to the File name field.
   - If you know the full pathname of the file to be programmed, enter the pathname in this field.
   - If you do not know the full pathname of the file to be programmed, click on the Browse button. This brings up a standard file-select dialog box: select the file and click on the OK button. This returns you to the main screen, entering the pathname in the File name field.
   - (If your only action for this Loader session will be uploading FLASH contents, you may leave the File name field blank.)
7. Make sure that the SYSTEM field shows the value CMB/EVB2080.
8. Use the FLASH area to configure the FLASH type, bus width, and size.

   The value in the Base Address field is automatic, according to the entry in the SYSTEM field. (Optionally, you may select the value <CUSTOM>, which brings up the Custom Address dialog box. Enter an appropriate address, then click on the dialog box OK button to return to the main screen.)

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

9. In the Communications area, use the Port field to specify the PC serial port, and use the Speed field to specify the communications rate. (The default rate is 19200 baud.)
10. To program FLASH memory, click on the Download button. As the software downloads the file you specified, a progress message appears in a Status dialog box. A Download successful message appears at the end of downloading: you are ready to use the code in FLASH memory.
    - If this is the first action of this SysDS Loader session, the software downloads an algorithm file before downloading the file you specified. A progress message appears during the downloading of this algorithm file.
    - If the software cannot find the algorithm file, an appropriate error message identifies the file. Click on the message's OK button to bring up a file-select dialog box, then use this dialog box to specify the location of the algorithm file. If necessary, recopy the file from the transmittal CD-ROM. Click on the OK button to resume programming FLASH memory.
    - The error message Unable to Validate Flash configuration indicates some problem with the programming. A likely such problem is that the chip select base address does not correspond to the configured chip select. Correct the problem, then click again on the Download button.
11. To upload FLASH memory contents to a file in your PC, click on the Upload button. This brings up the Upload To File dialog box, **Figure 3-2**:

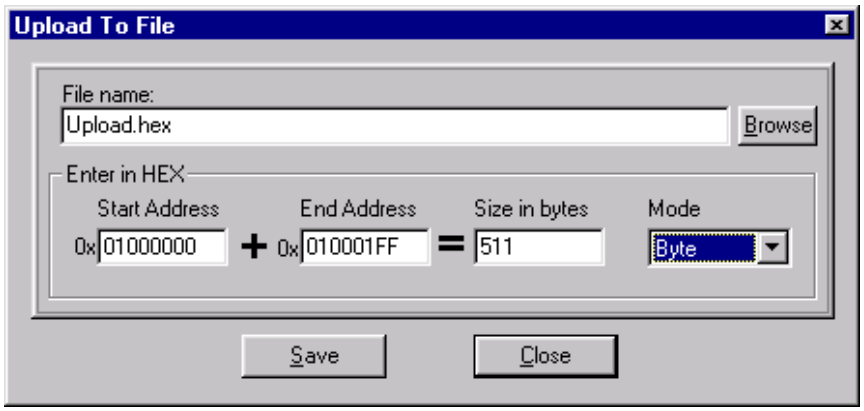


Figure 3-2 Upload To File Dialog Box

- Enter the name of the destination file. Optionally, click on the Browse button, to select a file via a standard file-select dialog box. (The SysDS Loader will append FLASH contents to the end of this PC file.)
- The Start Address field indicates the start of EVB2080 FLASH memory. The default address value corresponds to the value of the SYSTEM field of the main screen, but you may enter a different address, if appropriate.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

- The Size in Bytes field value corresponds to the value of the Size field of the main screen. (If appropriate, you may enter a different value.)
- The system determines the value of the End Address field from the Start Address and Size in bytes values.
- The default Mode field value is Byte.
- When the Upload To File dialog box shows appropriate values, click on the Save button. A progress message appears during uploading.

12. To verify that the contents of Flash memory match the selected download file, click on the Verify button. A progress message appears as verification begins. A Verify successful message appears at the end of verification.
    - If this is the first action of this SysDS Loader session, the software downloads an algorithm file before verifying FLASH. A progress message appears during the downloading of this algorithm file. (Should the software be unable to find the algorithm file, an appropriate error message appears, as explained under the program FLASH memory step, above.)
    - If verification fails, an error message specifies the location that did not have the expected contents.
    - To recover from a verification failure, try downloading Flash again, to replace the selected download file.
13. To view the contents of Flash memory, click on the Display button. This brings up the Display Flash/Ram display (**Figure 3-3**).

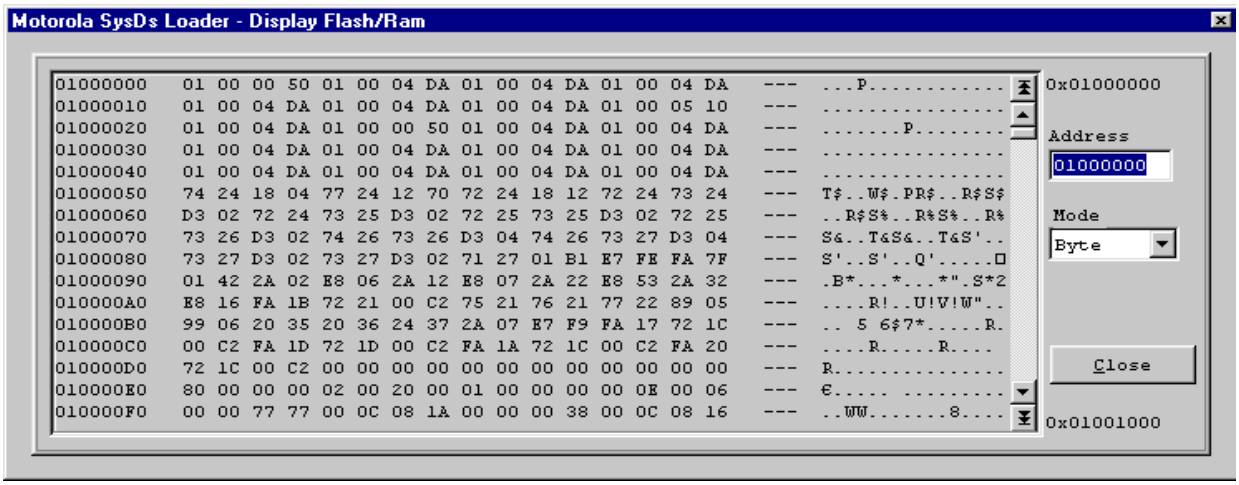


Figure 3-3 Display Flash/Ram Display

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

- If this is the first action of this Loader session, the software downloads an algorithm file before displaying FLASH contents. A progress message appears during the downloading of this algorithm file. (Should the software be unable to find the algorithm file, an appropriate error message appears, as explained under the program FLASH memory step, above.)
- The Address field shows the first address of the value display. One way to change the display is to enter a different address in this field.
- Another way to change the value display is to use the scroll bars.
- Use the Mode field to specify byte, half-word, or word values in the display.
- When you are done viewing the display, click on the Close button to return to the main screen.

14. To erase FLASH memory, click on the Erase FLASH button. The SysDS Loader erases all contents of the FLASH memory except for sectors 0 through 4 (which contain the system software). Erasing takes 20 to 30 seconds.

    If this is the first action of this SysDS Loader session, the software downloads an algorithm file before erasing FLASH. A progress message appears during the downloading of this algorithm file. (Should the software be unable to find the algorithm file, an appropriate error message appears, as explained under the program FLASH memory step, above.)

15. To erase a sector of FLASH memory, click on the Erase Sector button. This brings up the Flash Sector Number dialog box. Enter the number of the sector to be erased, then click on the OK button.

    - If this is the first action of this Loader session, the software downloads an algorithm file before erasing the FLASH sector. A progress message appears during the downloading of this algorithm file. (Should the software be unable to find the algorithm file, an appropriate error message appears, as explained under the program FLASH memory step, above.)
    - If you specify an of sectors 0 through 4, a message reminds you that the system software is in this sector. Buttons of the message box let you cancel the erasure or proceed with the erasure.

NOTE: *The target FLASH has 19 sectors, 0 through 18, of different sizes. Do not erase any of sectors 0 through 4, which contain system software, unless it is absolutely necessary.* ***Table 3-2****, at the end of this subsection, lists the contents and address ranges of each such sector.*

If you must erase any of these sectors, you subsequently can restore factory programming by clicking on the Restore System Software button. Paragraph 3.2.2 gives additional information about restoring system software.

Freescale Semiconductor, Inc.
For More Information On This Product,
Go to: www.freescale.com

16. To verify that a FLASH sector is blank, click on the Blank Check button. This brings up a dialog box that asks for a sector number. Enter the number of the sector to be blank checked, then click on the OK button.

    A message tells you the results of the blank check. (If the sector is not blank, you can erase the sector or try a different sector.)

17. To end your SysDS Loader session, merely close the main screen.

Table 3-2 shows contents of significant memory address ranges and sectors. Except in unusual circumstances, you should avoid programming any of these ranges not allocated for user code or user space.

Table 3-2 Memory Range/Sector Contents

Memory Type	Sector	Addresses	Contents
External FLASH	0	0x0100_0000 — 0x0100_07FF	Boot code
	0	0x0100_0800 — 0x0100_07FF	Programmer
	1, 2	0x0100_4000 — 0x0100_6FFF	Power-on and built-in selftests
	2	0x0100_7000 — 0x0100_7FFF	Device drivers
	3	0x0100_8000 — 0x0100_FFFF	Picobug
	4	0x0101_0000 — 0x0101_7FFF	ESL monitor
	5 — 18	0x0102_0000 — 0x010F_FFFF	User code
External SRAM		0x0200_0000 — 0x0200_03FF	Exception vector table
		0x0200_0400 — 0x0200_13FF	Stack
		0x0200_1400 — 0x0200_43FF	Programmer code and data
		0x0200_4400 — 0x0200_63FF	Boot/ESL/Picobug data
		0x0200_6400 — 0x0207_FFFF	User space

3.2.2 Restoring System Software

If you must overwrite FLASH-device sector 0, you subsequently may use the SysDS Loader to restore SysDS software to sector 0. To do so, follow these instructions:

1. Make sure that the communications rate is 19200 baud.
2. Move the header W2 jumper to pins 3 and 4.
3. Move the header W4 jumper to pins 1 and 2.
4. Set switch S3 subswitch 1 to BOOT INTERN; set S3 subswitch 2 to ROM EMU OFF.
5. Press switch S1, to reset the EVB2080.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

6. Click on the Restore System Software button of the main screen.
   - If the system software is in your current hard-disk directory, the SysDS Loader automatically restores factory programming to FLASH sectors 0 through 4. The main screen reappears to confirm successful programming.
   - If you receive a message that the system software does not exist, it may be because the software is in a different hard-disk directory. If so, make that directory the active one and click again on the Restore System Software button.
7. The main screen reappears to indicate successful restoration.

NOTE: *The SysDS Loader Verify button does not pertain to restored system software.*

8. End your SysDS loader session (by closing the main screen), and turn off or disconnect power from the EVB2080.
9. Return jumper header W2, jumper header W4, and switch S3 to their previous configurations.
10. Set switch S2 for the software you will use next.

3.3 MPB I/O Port Circuits

Circuits for the status LEDs and the general-purpose configuration switches share the processor MPB I/O port. Accordingly, the board makes use of some processor MPA port pins for control signals.

- The status LEDs DS1 through DS4 have alternate designations STAT3 through STAT0, respectively. The signal on MPA3 enables control for all four LEDs; the signals on pins MPB7 through MPB4 control individual LEDs.

  A low level on MPA3 lets a low level on an MPB pin light the corresponding LED. For example, if MPA3 and MPB6 both are low, LED DS2 (STAT2) lights. If the MPA3 signal is high or configured as an input, none of the LEDs can light.
- The general-purpose configuration switches are subswitches S2-1 through S2-4. The processor accepts settings of these switches as input data, on lines MPB0 through MPB3, if the MPA4 signal is configured as a low-level output. If a switch is ON, the MPB pin data is low (0); if a switch is OFF, the MPB pin data is high (1).

  If the MPB4 signal is high or configured as an input, switch S2 settings have no effect on the MPB pins.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 3-3 illustrates how EVB2080 functions share the MPB port. Note the LED and S2 subswitch dedicated control lines. Proper use of these control lines will prevent interference among these circuits.

Table 3-3 MPB, MPA Connections

GPIO Pins	LEDs	GP Switches
MPB7_ROW3	DS1 (STAT3)	
MPB6_ROW2	DS2 (STAT2)	
MPB5_ROW1	DS3 (STAT1)	
MPB4_ROW0	DS4 (STAT0)	
MPB3_COL3		S2-1
MPB2_COL2		S2-2
MPB1_COL1		S2-3
MPB0_COL0		S2-4
MPA4		S2 Enable
MPA3	LED Enable	
MPA2		
MPA1		
MPA0		

3.4 Accommodating Processor Differences

The resident MCU of your EVB2080 is a development-version MMC2080, in a 208-pin PQFP package. The EVB2080 lets you use this special MMC2080 device to simulate either a standard MMC2080 or a standard MMC2075.

The standard MCUs have fewer pins than does the development-version MMC2080: your hardware and code must take these differences into account. For more information about pin numbers and layout, see the chip user's manual.

Furthermore, the EVB2080, with its development-version MMC2080 MCU, offers more signals than either standard MCU. These signals are available to you during development, but your final code must avoid these signals, according to which MCU your product will use:

- The standard MMC2080 MCU lacks these development-extension signals:

  DSTAT[5] — DSTAT[0] DVLEB[1] DVLEB[0] DVLMX

  DVLSEL DVL[1] DVL[0] D[31] — D[16]

  HIGHZ PULL_EN TC[2] — TC[0] TEA

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

- The standard MMC2075 MCU lacks both the development-extension signals, and these FLEX signals:

  CLKOUT EXTS[1] EXTS[0] LOWBAT SYMCLK

  S0 S[7] — S[1]

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Section 4 Connector Information

This chapter consists of pin assignments and signal descriptions for EVB2080 connectors.

4.1 OnCE Connector (J27)

Connector J27, a 2-by-7-pin connector, conveys data and control signals to and from the OnCE control block. **Figure 4-1** and **Table 4-1** give the pin assignments and signal descriptions for this connector.

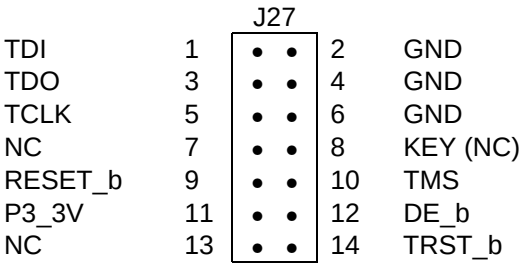


Figure 4-1 OnCE Connector J27 Pin Assignments

Table 4-1 OnCE Connector J27 Signal Descriptions

Pin	Mnemonic	Signal
1	TDI	TEST DATA INPUT – Data and command input line to the OnCE controller.
2, 4, 6	GND	GROUND
3	TDO	TEST DATA OUTPUT – Serial data output line from the OnCE controller.
5	TCLK	TEST CLOCK – Serial clock input line to the OnCE control block.
7, 8, 13	NC	No connection
9	RESET_b	RESET IN – Active-low input line to the OnCE controller, signalling a reset.
10	TMS	TEST MODE SELECT – Input signal that tells the OnCE control block to advance one mode state (of the cycle of mode states).
11	P3_3V	OPERATING VOLTAGE – Transmission line for +3.3-volt MCU operating power.
12	DE_b	DEBUG EVENT – Active-low debug-mode control line for the OnCE controller. An input signal from an external command controller makes the OnCE controller immediately enter debug mode. An output signal acknowledges debug-mode-entry to the external command controller.
14	TRST_b	TEST RESET – Active-low input line for an external reset signal to the OnCE controller.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

4.2 RS232 Connector (J26)

Connector J26, the RS232 connector, has DCE format. The diagram below shows the pin numbering of this connector. **Table 4-2** lists the pin assignments and signal directions for this connector.

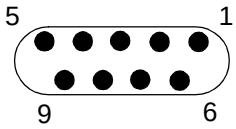


Table 4-2 RS232 Connector J26 Pin Assignments

Pin	Signal	Signal Direction
1	CD — Communication Detect (Held active on the EVB2080.)	Out
2	TXD — Transmitted Data	Out
3	RXD — Received Data	In
4	NC	—
5	GROUND	—
6	DSR — Data Set Ready (Held active on the EVB2080.)	Out
7	CTS — Clear to Send	In
8	RTS — Request to Send	Out
9	RI — Ring Indicator (Held inactive on the EVB2080.)	Out

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

4.3 Prototyping Connector Sites (J14, J16)

Board locations J14 and J16 are available for optional, user installation of Berg 69192-620 2-by-10-pin headers for wire wrapping, probing, or cabling to external prototype circuits. **Figure 4-2** and **Table 4-3** give the pin assignments and signal descriptions for such a connector installed at location J14. **Figure 4-3** and **Table 4-4** give the pin assignments and signal descriptions for such a connector installed at location J16.

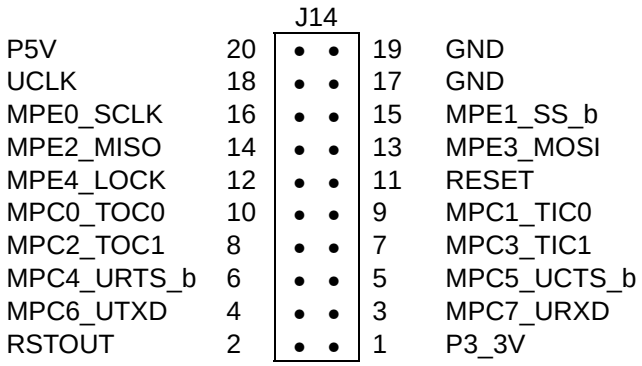


Figure 4-2 Connector Location J14 Pin Assignments

Table 4-3 Connector Location J14 Signal Descriptions

Pin	Mnemonic	Signal
20	P5V	5-volt power.
19, 17	GND	GROUND
18	UCLK	UART CLOCK — Clock signal for the UART.
16	MPE0_SCLK	SERIAL CLOCK — If SPI is enabled, the serial clock signal. If SPI is disabled, a general-purpose port E I/O signal.
15	MPE1_SS_b	SLAVE SELECT — If SPI is enabled, the slave polarity select signal. If SPI is disabled, a general-purpose port E I/O signal.
14	MPE2_MISO	MASTER IN/SLAVE OUT — If SPI is enabled, the data master-in/slave-out signal. If SPI is disabled, a general-purpose port E I/O signal.
13	MPE3_MOSI	MASTER OUT/SLAVE IN — If SPI is enabled, the data master-out/slave-in signal. If SPI is disabled, a general-purpose port E I/O signal.
12	MPE4_LOCK	EXTERNAL SYNTHESIZER LOCK — If the processor FSC bit is enabled, an input indicating that an external PLL has a lock. (Otherwise, a general-purpose port E I/O signal.)
11	RESET	RESET IN — Active-low input line that signals a reset.
10	MPC0_TOC0	TIMER 0 OUTPUT CAPTURE — An output signal. (Otherwise, a port C general-purpose I/O signal.)
9	MPC1_TIC0	TIMER 0 INTPUT CAPTURE — An input signal. (Otherwise, a port C general-purpose I/O signal.)

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 4-3 Connector Location J14 Signal Descriptions (Continued)

Pin	Mnemonic	Signal
8	MPC2_TOC1	TIMER 1 OUTPUT CAPTURE — An output signal. (Otherwise, a port C general-purpose I/O signal.)
7	MPC3_TIC1	TIMER 1 INTPUT CAPTURE — An input signal. (Otherwise, a port C general-purpose I/O signal.)
6	MPC4_URTS_b	REQUEST TO SEND — Active-low output signal. (Otherwise, a port C general-purpose I/O signal.)
5	MPC5_UCTS_b	CLEAR TO SEND — Active-low input signal. (Otherwise, a port C general-purpose I/O signal.)
4	MPC6_UTXD	TRANSMIT SERIAL DATA — Output signal. (Otherwise, a port C general-purpose I/O signal.)
3	MPC7_URXD	RECEIVE SERIAL DATA — Input signal. (Otherwise, a port C general-purpose I/O signal.)
2	RSTOUT	RESET OUT – Active-low output signal that resets external components. Activation of any internal reset sources asserts this line.
1	P3_3V	OPERATING VOLTAGE – Transmission line for +3.3-volt MCU operating power.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

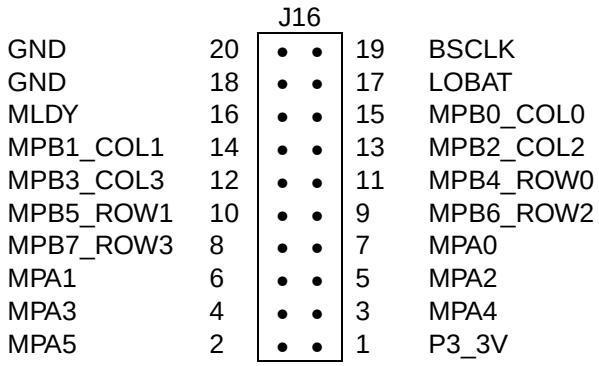


Figure 4-3 Connector Location J16 Pin Assignments

Table 4-4 Connector Location J16 Signal Descriptions

Pin	Mnemonic	Signal
20, 18	GND	GROUND
19	BSCLK	BUS CLOCK — Clock signal for the external bus.
17	LOBAT	LOW BATTERY – Signal that simulates a low-battery condition.
16	MLDY	MELODY — Melody generator output waveform signal.
15 — 12	MPB0_COL0 — MPB3_COL3	COLUMN SELECT (lines 3—0) — Open-drain output, column-select keypad signals. (Otherwise, general-purpose port B I/O signals 3—0.)
11 — 8	MPB4_ROW0 — MPB7_ROW3	ROW SELECT (lines 3—0) — Input, row-select signals. (Otherwise, general-purpose port B I/O signals 7—4.)
7 — 2	MPA0 — MPA5	PROGRAMMABLE PINS — Port A pins, individually programmable as input, output, or external interrupt.
1	P3_3V	OPERATING VOLTAGE – Transmission line for +3.3-volt MCU operating power.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Index

C

chip select headers (W2, W4) 17
components, configuring 13–21
computer system connections 22
configuration 13–25
configuration table 13–16
configuring components 13–21
connections, computer system 22
connector information 41–45
connector pin assignments
    connector site J14 43
    connector site J16 45
    OnCE connector J27 41
    RS232 connector J26 42
connector signal descriptions
    connector site J14 43
    connector site J16 45
    OnCE connector J27 41

D

debugging embedded code 27–32
downloading to FLASH memory 32–38

E

EVB2080
    layout 10, 11
    specifications 12

F

features 9
FLASH data bus access header (W1) 16

G

GNU source-level debugger 32

I

introduction 9–12

L

layout 10, 11

M

memory byte state switch (S3) 21
memory maps 24
MPB I/O port circuits 38, 39
mystery connector sites 45

O

OnCE connector 41
operation 27–40

P

Picobug debug monitor
    commands 28
    sample session 29–31
    using 27–31
pin assignments
    connector site J14 43
    connector site J16 45
    OnCE connector J27 41
    RS232 connector J26 42
processor differences 39
processor power connect headers (W5-W8) 18, 19
prototyping area 25
prototyping connector sites 43

R

requirements, system/user 10
RS232 connector 42

S

self-test 22, 23
setting components
    chip select headers (W2, W4) 17
    FLASH data bus access header (W1) 16
    memory byte state switch (S3) 21
    processor power connect headers (W5-W8) 18, 19
    software select switch (S2) 20
    SRAM data bus access header (W3) 18
signal descriptions
    connector site J14 43
    connector site J16 45
    OnCE connector J27 41
software select switch (S2) 20
specifications 12

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SRAM data bus access header (W3) 18
status LEDs 38
SysDS loader
    restoring system software 37
    steps 32–37
system requirements 10
system software, restoring 37

U

user requirements 10

Freescale Semiconductor, Inc.

Revision History

Revision Number	Date	Author	Summary of Changes
Original	Dec 1999	MTC DDOC	Original document.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Revision History

Freescale Semiconductor, Inc.

This manual is a product of the Motorola M•CORE Technology Center Design Documentation team. Technical writing, illustration, and production editing performed with Adobe® Framemaker® running on multiple platforms. Printed by Ken Cook, Inc. in Milwaukee, Wisconsin.